(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 January 2001 (11.01.2001)

PCT

(10) International Publication Number
WO 01/02973 A1

(51) International Patent Classification[7]: G06F 15/16, 15/82, 17/60

(21) International Application Number: PCT/US00/17921

(22) International Filing Date: 28 June 2000 (28.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

09/347,056	2 July 1999 (02.07.1999)	US
09/347,055	2 July 1999 (02.07.1999)	US
09/347,434	2 July 1999 (02.07.1999)	US
09/347,057	2 July 1999 (02.07.1999)	US
09/347,058	2 July 1999 (02.07.1999)	US
09/507,285	18 February 2000 (18.02.2000)	US

(71) Applicant: COVAD COMMUNICATIONS GROUP, INC. [US/US]; 2330 Central Expressway, Santa Clara, CA 95050 (US).

(72) Inventor: SAAVEDRA, Ralph; 2533 Aranda Drive, San Ramon, CA 94583 (US).

(74) Agent: KUO, Jung-Hua; P.O. Box 3275, Los Altos, CA 94024 (US).

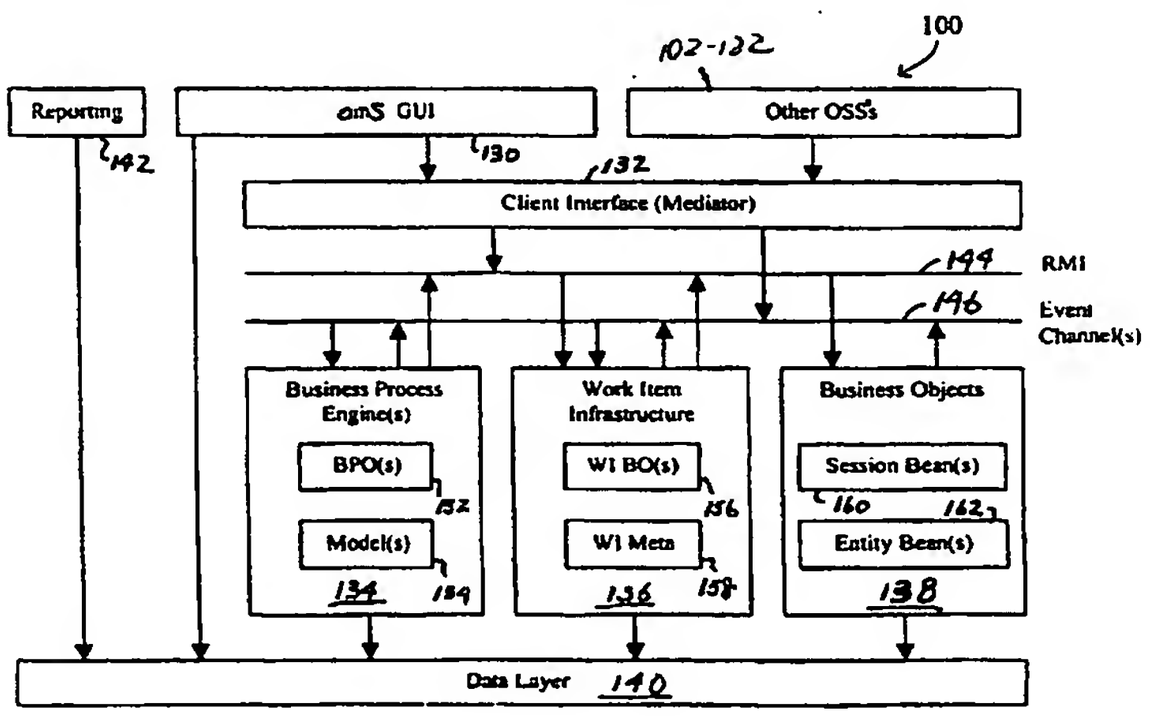
(81) Designated State *(national)*: SG.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: PROCESS FULFILLMENT SYSTEMS AND METHODS USING DISTRIBUTED WORKFLOW MANAGEMENT ARCHITECTURE



(57) Abstract: Systems and methods for order fulfillment using distributed workflow management architecture are disclosed. The system generally comprises a business objects module (138) defining business rules relating to the process, a business processes module (134) defining state models of the process, each state model defining states and state transitions, and a work item infrastructure module (136) containing work item business objects (156) and work item meta data (158). The business process module (134) is preferably configured to cause generation of work items in the work item infrastructure (136) on which work item actions can be performed and the performance of work items actions causes transitions within the state model in the business processes module (134) and causes updating of the business objects module (138).

WO 01/02973 A1

PROCESS FULFILLMENT SYSTEMS AND METHODS USING DISTRIBUTED WORKFLOW MANAGEMENT ARCHITECTURE

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of co-pending U.S. Patent Application Serial Nos.:

(1) 09/347,055, entitled "Securing Local Loops for Providing High Bandwidth Connections";

(2) 09/347,056, entitled "Processing Orders for High Bandwidth Connections";

(3) 09/347,057, entitled "Provisioning Virtual Circuits for High Bandwidth Connections";

(4) 09/347,058, entitled "Determination of DSL-Based Services Possible (Feasible) to a User Location"; and

(5) 09/347,434, entitled "Rolling Out High Bandwidth Connection Services in Geographical Areas Covering Several Central Offices," all filed on July 2, 1999, each of which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to process fulfillment systems and methods using distributed workflow management architecture. More specifically, systems and methods for distributed work flow management, such as an operational support system (OSS) to facilitate the service order fulfillment by a provider of digital subscriber line (DSL) or other high bandwidth connections to users at remote locations, are disclosed.

2. Description of Related Art

With the increasing popularity of the Internet and telecommuting, more and more home and business users order and subscribe to high bandwidth connections to Internet service providers (ISP) and/or corporate networks. Such high bandwidth connections may be provided by Integrated Services Digital Network (ISDN) providers, cable providers, or DSL providers, for example. Examples of DSL providers include incumbent local exchange carriers (ILECs) such as Pacific Bell (PacBell) of California and competitive local exchange carriers (CLECs) such as Covad Communications Group, Inc., assignee of the subject patent application. ILECs and CLECs are commonly referred to as LECs herein.

Efficient and effective delivery of service, modifications to the service order, and addressing of exceptions during delivery and modifications often require the coordination of several tasks and the use of related information by various service provider personnel and/or contractors. The tasks may be performed in numerous stages of the service delivery process, the service modification process, and the exceptions addressing process. The status of the tasks in a given stage also may impact other tasks in that stage and/or tasks in other stages. For example, in an initial set-up stage, a service provider may need to setup several types of equipment at the customer site and/or at the Central Office (CO) prior to delivery of the service ordered by or for the customer.

It would be thus desirable to provide an order management system for fulfilling service orders while coordinating among the tasks and the interchange of information. Preferably, the order management system fulfills various orders from the customers such as new installation orders, changes or supplements to or cancellation of an order-in-progress prior to service being turned up, a modification order of a service already in place, and/or disconnect order to delete the services. In addition, with an increasing number of service orders, it would be desirable to provide an order management system that facilitates efficient and effective coordination of tasks and usage of personnel time, and require minimal personnel training while providing useful feedback information.

SUMMARY OF THE INVENTION

Systems and methods for process fulfillment using distributed workflow management architecture are disclosed. For example, the systems and methods for process fulfillment may be implemented in an operational support system (OSS) to facilitate the service order fulfillment by a provider of digital subscriber line (DSL) or other high bandwidth connections to users at remote locations. It should be appreciated that the present invention can be implemented in numerous ways, including as a process, an apparatus, a system, a device, a method, or a computer readable medium such as a computer readable storage medium or a computer network wherein program instructions are sent over optical or electronic communication lines. Several inventive embodiments of the present invention are described below.

A distributed workflow system for fulfillment of a process generally comprises a business objects module defining business rules relating to the process, a business processes module defining state models of the process, each state model defining states and state transitions, and a work item infrastructure module containing work item business

objects and work item meta data. The business process module is preferably configured to cause generation of work items in the work item infrastructure on which work item actions can be performed and the performance of work items actions causes transition within the state model in the business processes module and causes updating of the business objects module. The business processes module may include business process objects for creating the work items in the work item infrastructure. The system may further comprise a data layer for storing data obtained by at least one of modules and a reporting module for processing data layer data and for report generation.

The system may further comprise a remote method invocation bus for invoking business methods and for enabling an external system to activate components in the business objects module and an events channel, wherein the three modules are adapted to post events of the process on the events channel. The business processes module and the work item infrastructure module may selectively subscribe to events posted on the events channel.

The business objects module may comprise at least one entity bean and at least one session bean, where the session bean is adapted to invoke business logic and to enforce the business rules. The systems may include a meta session bean, an order session bean, a work items session bean, and a user session bean.

The business objects module may also include data relating to roles and profiles assigned to end users such that assignment of each work item action to the end user is based upon the role and profile specified by the work item meta data and the role and profile of the end user.

The method for process fulfillment using a distributed workflow architecture generally comprises generating work items by a business process engine on a work item infrastructure module, the work item infrastructure module containing work item business objects and work item meta data, transitioning within a state model of the process defined in the business processes module upon performance of a work item action associated with one of the generated work items, the state model defining states and state transitions, updating a business objects module upon performance of the work item action, the business objects module defining business rules relating to the process, and repeating the generating, transitions, and updating for each state of the state model until disposition of the process.

As noted, the systems and methods for process fulfillment may be implemented in an operational support system (OSS) to facilitate service order fulfillment by a provider of digital subscriber line (DSL) to users at remote locations. Specifically, the order

management system may fulfill various orders from customers. Examples of customer orders include new installation orders, changes or supplements to or cancellation of an order-in-progress prior to service being turned up, a modification order of a service already in place, and/or disconnect order to delete the service.

The order management system facilitates in management of the various tasks and the interchange of information amongst the tasks. The order management system also allows the addition, modification, or deletion of various tasks or portions of the order fulfillment process. For example, it may be desirable to modify the performance of a single loop test on the firm order commitment (FOC) date provided by the ILEC for loop delivery of the loop ordered by the CLEC to performing three loop tests at the rate of once per day starting one day prior to the FOC date and ending one day after the FOC date. The order in which the tasks are to be performed may also be modified within the order management system.

The order management system also allows the addition, modification, or deletion of various tasks or portions of the order fulfillment process. For example, it may be desirable to modify the performance of a single loop test on the firm order commitment (FOC) date provided by the ILEC for loop delivery of the loop ordered by the CLEC to performing three loop tests at the rate of once per day starting one day prior to the FOC date and ending one day after the FOC date.

Furthermore, with the increase in service orders, the order management system is readily scalable to accommodate the increased demand. In addition, as more personnel and supervisors become involved in the order fulfillment process, increasing the organizational complexity, the order management system facilitates the efficient and effective coordination of tasks and usage of personnel time and the prioritization of the active tasks while requiring minimal personnel training. With the increase in service fulfillment, the amount of information required and generated similarly increases. The order management system optionally provides useful feedback information by maintaining work item histories, collecting data and generating reports to provide information regarding the effectiveness of certain processes, identifying observable trends, identifying bottlenecks and/or frequency of exceptions, enabling effective personnel review and review of the ILEC and the customer performance in giving information, for example.

With the occurrence of exceptions during the order fulfillment process and the variances in the completion time of each task in the order fulfillment process, the order fulfillment system maintains both relative time for a given work item and absolute time from the time of the service order. The order fulfillment system may then automatically

prioritize various tasks such that the personnel may fulfill service requests more effectively.

The order fulfillment system also allows the addition, refinement, or modification of the steps themselves, the order of the steps, and the attributes of the steps, such as the time frame within which the task is to be completed, the time frame within which work is to begin for the given task, or the role of the personnel who bears the responsibility to perform the task.

These and other features and advantages of the process fulfillment systems and methods using distributed workflow architecture will be presented in more detail in the following detailed description and the accompanying figures which illustrate by way of examples the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like elements, and in which:

FIG. 1 is a block diagram illustrating interactions of an order management system with various other operational support systems (OSS's) external to the order management system which, in combination, support various DSL service fulfillment processes;

FIG. 2 is a block diagram illustrating a functional architectural overview of the order management system;

FIG. 3 is a block diagram illustrating one embodiment of the technical hardware and software architecture of the order management system ;

FIG. 4 is a block diagram illustrating the interactions of the order management system GUI with various session beans and meta data;

FIG. 5 illustrates an example of an assignment plan mapping end users and profiles;

FIG. 6 is a flow chart illustrating a process for fulfilling a single service order utilizing the order management system in the absence of exceptions;

FIG. 7 is a state diagram of a state model/business process illustrating the work items associated with a service installation order business process in the absence of exceptions;

FIG. 8 is a state diagram of a state model/business process or a portion of a state model/business process illustrating the work items associated with a loop not delivered due to busy pair exception;

FIG. 9A illustrates an example of a GUI for specifying the criteria for generating reports from the data maintained by the order management system;

FIG. 9B illustrates an example of a workflow summary report generated using the data maintained by the order management system;

FIG. 10 illustrates an example of a computer system that can be utilized with the various embodiments of method and processing described herein; and

FIG. 11 illustrates a system block diagram of the computer system of **FIG. 10**.

DESCRIPTION OF SPECIFIC EMBODIMENTS

Systems and methods for process fulfillment using distributed workflow management architecture are disclosed. The following description is presented to enable any person skilled in the art to make and use the invention. Descriptions of specific embodiments and applications are provided only as examples and various modifications will be readily apparent to those skilled in the art. The general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is to be accorded the widest scope encompassing numerous alternatives, modifications and equivalents consistent with the principles and features disclosed herein. For purpose of clarity, details relating to technical material that is known in the technical fields related to the invention may not be described or shown in detail so as not to unnecessarily obscure the description of various embodiments.

Overview of Distributed Workflow Management System and Method

A general overview of the overall order management system and method will be presented with reference to **FIGS. 1** and **2**. For purposes of clarity only, the discussion herein is generally directed to a system that may be utilized by a DSL service provider such as a CLEC in fulfilling service orders in order to provide DSL services to remote customers, for example. In addition, the service provider that utilizes the workflow management system and method is referred to herein as a client of the order management system. Each personnel such as an employee or a contractor of the service provider is

referred to herein as individual end users of the order management system. For purposes of clarity, a party submitting a service order is referred to herein as a customer.

FIG. 1 is a block diagram illustrating the interactions of the order management system 100 with various other operational support systems (OSS's) external to the order management system which, in combination, support various DSL service fulfillment processes. Although the order management system 100 is shown to be the only system in communication with each of the external OSS's, it is to be understood that each external OSS may be in communication with any or all of the other OSS's.

As shown, the order management system (OMS) 100 may interact with a customer facing interface/extensible markup language (CFI/XML) system 102, an ILEC gateway such as an electronic data interchange (EDI) system 104, an inventory management system 106, a high capacity circuit system 108, a provisioning system 110, a loop test system 112, the workflow management system (WFMS) 114, a client premise equipment (CPE) wizard 116, a billing system 118, a Trouble Ticketing (TT) system 120, and a Fault Management (FM) management system 122. The interactions between the order management system 100 and each of the OSS's 102-122 as well as the functions of each of these systems are described in more detail below.

The CFI/XML system 102 allows the customer to place orders, such as service installation orders, disconnection of in-place service orders, changes to and cancellations of any type of orders, and service change supplemental orders to the order management system 100. The CFI/XML system 102 optionally allows the customer to save complete or partially complete orders prior to submitting the order to the order management system 100, as well as obtain order, circuit, and/or service status for its order(s).

The order management system 100 may drive the ILEC preordering and ordering functions through the ILEC EDI 104. In particular, the order management system 100 may obtain loop orders for loops that require placement, such as install, move, disconnect, cancel, change orders, as well as obtain loop order status reports via the ILEC EDI gateway 104.

The order management system 100 may also communicate with the inventory management system 106 for making requests. Requests to the inventory management system 106 may include reserve, release, crossConnect, release crossConnect, and convert pairs requests, as well as reserve, release, and change customer PVC requests, for example. In addition, the order management system 100 may obtain the status of customer backhaul circuits via the high capacity circuit system 108.

The order management system 100 may make provisioning and/or deprovisioning requests to the provisioning system 110 for services that require provisioning and/or deprovisioning and record the (de)provisioning results as provided by the provisioning system 110. The order management system 100 may also make loops test requests to the loop test system 112 for loops to be tested and record the loop test results as provided by the loop test system 112. Further, the order management system 100 may provide work items to the work force management system (WFMS) 114 for assignment and/or scheduling of tasks or work items and to obtain status change of work items.

The order management system 100 may also request a download of CPE configuration information required for a customer installation from the CPE wizard 116. In addition, the order management system 100 may notify the billing system of all billable events, such as installations, changes to services, and disconnection of services, as well as customer service provider switches, cancellation, service change, and/or a no show at the customer site, for example.

Although not shown, various other external OSS's may be coupled to and in communication with the order management system 100 and/or any of the external OSS's shown and described. For example, a service assurance system may be coupled to the order management system 100 so as to retrieve customer information associated with circuits and services as well as the state of circuits and services from the order management system. As another example, the order management system 100 may retrieve service configuration information, such as technology mappings, service provisioning profiles, and billing profiles, from a service management system.

Architectural Overview of the Distributed Workflow Management System and Method 100

FIG. 2 is a block diagram illustrating a functional architectural overview of the distributed workflow management system 100. The distributed workflow management system 100 manages orders or processes and corresponding tasks or work items using a distributed workflow architecture. Some of the basic functions of the order management system 100 includes managing and driving order processes toward fulfillment or completion as well as collecting and maintaining historical order and order fulfillment processing data for effectively and efficiently reporting status and analysis of the fulfillment processes.

As shown, the order management system 100 generally comprises an order management system graphical user interface (OMS GUI) 130, a client interface/mediator

132, one or more business process or state engines 134, a work item infrastructure 136, business objects 138, a data layer 140, and a reporting module 142. The business process engine 134 includes business process object(s) (BPO's) 152 and business processes or state model(s) 154. The work item infrastructure 136 includes work item business object(s) 156 and work item meta data 158. In addition, the business objects 138 include session bean(s) 160 and entity bean(s) 162.

The order management system 100 further comprises a remote method invocation (RMI) bus 144 and an event channel(s) bus 146. The RMI bus 144 and the event channel bus 146, along with the client interface/mediator 132, facilitate communications among the GUI 130, the other OSS's 102-122, the business process engine 134, the work item infrastructure 136, and the business objects 138. In particular, events are created by the system 100 and are published and reside on the event channels 146. Further, external systems 102-122 only communicate with the work item infrastructure server 136 and the business objects server 138 and do not communicate directly with the business process engines 134.

Each of the business process engines 134, the work item infrastructure 136, and the business objects 138 as well as the GUI 130 and the reporting module 142 communicates with the data layer 140. The GUI 130, along with other OSS's 102-122, communicates with the business process engine 134, the work item infrastructure 136, and the business objects 138 via the client interface/mediator 132 and/or the RMI and the event channel buses 144, 146.

The reporting module 142 preferably directly accesses the data layer 140 to create reports for end users. Reporting and business management requirements may include the collection of data and metrics which allows the analysis and improvement of business processes and the performance of end users, suppliers, and customers. The data layer 142 provides a combination of on-line transaction processing (OLTP) and data warehouse access.

The distributed workflow order management system and associated methods for processes described herein utilize a generic reusable architecture comprising reusable components. In particular, the distributed workflow order management system is generally separated into three components: business objects, business process definition, and meta data in the form of a work item infrastructure module or server.

The business objects module contains static business logic and/or rules. For example, the business objects module may store information regarding fundamental

business rules and facilitate in keeping those fundamental business rules in check, i.e., maintain data integrity. Examples of fundamental business rules in the context of a DSL order fulfillment system may include the requiring an Internet Protocol (IP) address when creating an order, a maximum of one active loop at a time, a maximum of one twisted pair assigned to a work item at a time. These static business logic generally do not contain information regarding the business process.

The business process definition module or server generally do not contain information regarding the fundamental business rules but rather contains the processes to be followed, e.g., the state models and the corresponding state transitions. For example, again in the context of a DSL order fulfillment system, the business process definition module may specify that an installation cannot be scheduled until receipt of a FOC date. As is evident, the business process definition module defines the dependencies between actions.

The work item infrastructure module serves to intermediate or bind together the business process definition module and the business objects module. Specifically, the business process module causes work items to be created in the work item infrastructure such that events are posted on the events channel. When actions on work items are performed, the business objects module is updated and the business process module or the state model is driven forward.

The work item infrastructure module contains the work item meta data. Meta data refers to descriptive data regarding work items and state models. For example, the meta data may define what information, i.e., action data, to collect for a given work item, what business object method to invoke, what event(s) to publish, and/or what exception flow to raise or create. In other words, the work items created in the work item infrastructure have characteristics described by meta data.

The architecture and components of the order management system and associated methods described herein simplify process definition as well as adoption of process changes. The order management system can be adapted and utilized for any suitable purpose, such as for workflow management associated with DSL service order fulfillment by an LEC. The order management system can also be adapted for other uses, such as circuit ordering, trouble ticket applications, diagnosis and resolution of problems issues. Generally, the order management system is particularly suitable for use in any operation intensive, i.e., process centric, processes.

Because of the architecture of the distributed workflow order management system, the order management system may be easily adapted and reconfigured for different

applications, such as modifying meta data, state models, and/or business rules or logic. In particular, by creating, modifying, or deleting applicable state models in the business process engine, state models/business processes may be customized, modified or created. For example, changes in the order or flow of work items for fulfilling a service installation order may be implemented by modifying the state model for the corresponding state model/business process and/or modifying the meta data for the associated work items.

In addition, because work item characteristics are also defined by meta data, work items can be added to one or more state models, modified, or deleted by modifying the meta data of the work item or of the state model. Work item types may also be created (such as to account for exceptions), modified or deleted. The distributed workflow management system 100 may be utilized for client service order fulfillment to support the LEC's core service fulfillment requirements. Such core service fulfillment requirements may include the processing of service installation orders, disconnection of in-place service orders, changes to and cancellations of any order type, and associated exceptions. Exception processes may include ILEC resolution, Technical Assistance Center (TAC) resolution, field jeopardy, missed FOC date, and/or facility issues, for example.

The use of the distributed workflow system facilitates in reducing the work required to implement the creation and change of business processes. The distributed workflow system can be scalable so as to meet increasing performance and capacity requirements. The distributed workflow system also facilitates in maximizing user efficiency with minimal user training and in collecting data and metrics to allow analysis and improvement of business processes and the performance of LEC employees, suppliers, customer, for example.

Business Processes/State Models 154

As noted, the business process or state engine 134 contains state models 154. A business process or a state model describes the order fulfillment process of a given order, including that of any associated exceptions. Multiple independent state models having states, actions, and state transitions may be maintained by the business processes/state models 154. A state model 154 is preferably defined for each order to be processed and fulfilled such as service installation orders, disconnection of in-place service orders, changes to and cancellations of any order type, including any associated exceptions such as failures and/or errors. More specifically, examples of DSL service fulfillment orders for which business processes or state models 154 are provided in the business process engine 134 include installation order, cancel installation order, change service before installation

with or without technology change, change service during installation with or without technology change, disconnect order, cancel disconnect order, change order with or without technology change, cancel change order, and cancel customer switch order.

Any or all of these service order types may also include associated exceptions. Alternatively, an exception process may be defined by a separate state model 154. Examples of exceptions for which state models 154 are provided in the business process engine 134 include failed loop test exception, failed dispatch exception, loop not delivered due to no access or busy pair exception, loop order quality issue exception, downgrade exception, service change failed dispatch exception, failed provider switch exception, short or long term facility issue exception, technically not feasible facility issue exception, inventory/provisioning issue exception.

The state model 154 defines the states as well as the state transitions of the fulfillment process for a corresponding order. An order may be in one or more states at a given time and/or be in one or more state model such as with parallel flows. For example, exceptions can be part of the state model for the process with which it is associated or be part of a separate state model. Fulfillment of the corresponding order requires the transitioning from a start state through some set of sets to a final state. In the absence of exceptions, an order process transitions through a subset of the states, the subset of states being referred to as the normal states. All other states for the corresponding order are referred to as exception states which may only be entered with the occurrence of one or more exceptions.

Work Items and Corresponding Work Item Actions

Within the order management system 100, zero or more work items may be created during each state of the order process and the current work items depend upon the current state of the order process. Work item actions are optionally required to complete a given work item, may trigger the creation of work items, and may be invoked by the GUI 130 or the business objects 138. Actions on a work item are performed by calls to the work item infrastructure 136. Thus, various actions are typically required to fulfill a given service order so as to move the state from the start state to the final state.

The work items required in a normal state are referred to as normal work items while all other work items required, i.e., only through at least one exception state, are referred to as exception work items. The particular work item action(s) required by a work item may be completed automatically by some system external to the order management system and/or manually by the end user.

Typically, each business process engine 134 controls multiple corresponding BPO's 152. The business process engine 134 controls the lifecycle of and instantiates the corresponding BPO 152 which executes the associated state model 154. The BPO 152 creates work items in the work item infrastructure 136 to drive transitions in the applicable state model 154.

Meta Data

Meta data refers to descriptive data regarding work items and state models, for example and contains various key information. For example, the meta data may define what information, i.e., action data, to collect for a given work item, what business object method to invoke, what event(s) to publish, and/or what exception flow to raise or create.

The work items created in the work item infrastructure 136 have characteristics described by meta data. Work items meta data may include description of how a given work item is to be presented and the possible actions for the given work item. Meta data may include any of several fields. For example, the meta data may specify the role of the end user who should or may perform the task for the work item, such as any or certain specific end user(s) in NOC, TAC, and/or field operations. The meta data may specify when the task for the work item is to be performed, such as immediately or after a certain period of time to allow a loop order to process, for example. The meta data may also specify the time frame for completion and/or escalation criteria. For example, depending on whether the work item is on or off track with respect to a relative time as measured from the start of the work item and/or with respect to an absolute time as measured from the time of the order, the meta data may specify escalation level of normal, moderate, high, or another level of severity.

With respect to the state models, the meta data preferably also includes information regarding the state model such as what actions and/or events cause transitions from one state to another. In addition, the meta data may specify all possible transitions of each state model and all possible work items.

The GUI 130 optionally loads meta data from the work item infrastructure 136 at the startup of the order management system and caches the meta data. As shown in **FIG. 2**, the GUI 130 preferably directly accesses the data layer 140, particularly for large sets of read-only data.

Invoking Business Methods to Complete Work Item Actions

The work item actions may be invoked by the GUI 130 or the business objects 138. A work item action is invoked to complete the work item, to publish events to the business process engine 134, to cause transitions in the applicable state model 154, and to update work item status.

A work item action may be achieved or completed by invoking business methods in the work item infrastructure 136. The other OSS's 102-122 may also invoke business methods on the work item infrastructure 136. The work item infrastructure 136 ensures the transactional integrity of work item actions.

When an end user invokes an action on a work item business object 156, the GUI 130 generates a list of input parameters by utilizing the work item meta data 158. The GUI 130 sends work item identifiers and the input parameters to the meta session bean 160 to invoke a transaction on the business objects 138.

Technical Hardware and Software Architecture

FIG. 3 is a block diagram illustrating one embodiment of the technical hardware and software architecture of the order management system 100. As shown, local and remote clients 172A, 172B are in communication with application servers 174, which are in turn in communication with a database server 176. The local and remote clients 172A, 172B run the order management system GUI application software while the application servers 174 run the business process engines, the work item infrastructure server, the business objects servers, the event channels bus, as well as the RMI bus, which are described above with reference to **FIG. 2**. In addition, the database server 176 maintains the database.

Order Management System GUI Interactions

FIG. 4 is a block diagram illustrating the interactions of the GUI 130 with various session beans 160 and with the meta data. Session beans 160 provide the external interface of the business object server (shown in **FIG. 2**) for invoking business logic. All requests from external systems are preferably submitted through the session bean interface. A session bean enforces the business rules and manipulates the business data objects to perform a business operation.

As shown, the GUI 130 interfaces with the user session bean 160A, the work items session bean 160B, and the order session bean 160C. Although these four session beans

160 are shown, the session beans 160 may additionally or alternatively include various other session beans.

Various ways for storing meta data may be implemented. For example, an XML document 164 may be available to the GUI in the form of an ASCII file via a meta data interface 166 across an application program interface (API) 168. Alternatively, although not shown, hard coded Java Property Description files can be auto-generated from the meta data representation in the database, compiled and available as Java object tree at runtime.

The GUI 130 interfaces with the user session bean 160A to verify user login and to update and query profile, role, assignment, and/or user-related information. The GUI 130 interfaces with the work item session bean 160B to load work items listed based on the user profile stored in the order management system and to perform work item transitions. The work items list may be updated using status events and assignment event, preferably in real time.

The GUI 130 listens for user management-related events on channel 170A from the user session bean 160A and for status events on channel 170B from the work items session bean 160B. In particular, the user management-related events from the user session bean 160A may include user changed event when there is a change in a user object, role changed event when there is a change in a role object, role profile changed event when there is a change in a profile object, assignment plan changed event when there is a change in an assignment plan, and/or a force logout event to notify the GUI 130 to force the logout of a user. Further, the status events from the work items session bean 160B may include update status event when the status of a work item changes, update severity event when the severity of a work item change, send message to user event when an asynchronous message is delivered from one user to another user, order status dirty event when an order status has changed, and/or order dirty event when some attribute of an order has changed.

The GUI 130 also interfaces with the order session bean 160C to retrieve the order lists and order details and to update the order. The order session bean 160C provides the external interface for invoking work item actions using business logic. Lastly, as discussed above, the GUI 130 sends work item identifiers and the input parameters to the meta session bean 160D to invoke a transaction on the business objects.

In addition to session beans 160, the business objects 138 also contains entity beans (shown in **FIG. 2**). Methods may be invoked on entity beans, such as order entity beans, to create and maintain persistent information. Preferably, entity beans are not directly

exposed to the external systems and all business rules are enforced by the corresponding session bean.

As shown in **FIG. 3**, channels 170A, 170B utilize Enterprise Java Bean (EJB) objects. EJB objects are utilized for the remote invocation of business objects, either directly or through a session bean, such as the user session bean 160A and the work items session bean 160B. As is known in the art, EJB is typically used to build Java applications that run in a server by using a "container" layer that provides common functions such as security and transaction support and delivers a consistent interface to the applications regardless of the type of server.

The RMI bus 144 (shown in **FIG. 2**) may be utilized to invoke methods against EJB objects. The RMI bus 144 enables a client program to activate components in the business objects server 138. On the other hand, the event channel 146 provides guaranteed delivery of events and allow publication/subscription to events which drive the business process engines 134.

End Users, Profiles, Roles, Assignment Plan

As discussed above, each personnel such as an employee or a contractor of the service provider is referred to herein as an end user of the order management system. Each end user is associated with one or more roles. Typically, roles correspond to departments within an organization or a company. Example of roles may include software engineering, operations, sales, marketing, finance, call center, customer care operations, service delivery operations, NOC, field operators (dispatch, dispatch supervisor), TAC, information technology (IT). A role may include one or more profiles. Typically, profiles correspond to job functions within a particular department. For example, the service delivery operations role may include an ILEC specialist profile, a jeopardy analyst profile, and a service delivery supervisor profile. As another example, the field operators role may include a dispatch profile and a dispatch supervisor profile.

Each role is preferably defined with a role name, a description, a supervisor, create time, and status (e.g., active or inactive). A role may be associated with specified end users, work item types, and/or profiles. Each end user is preferably associated with one or more roles that determine which work item type (described below) on which a user may perform work to complete the work item. In other words, the meta data of each work item specifies the role(s) of the end users who may perform the tasks associated with the work item.

A profile is associated with a single role and one or more profiles may be associated with each role. Profiles may be used to allow one or more end users with the same role to focus on different aspects of the role. Profiles may define the prioritization of work items corresponding to the role according to the specified sorting criteria, for example. Examples of sorting criteria include escalation level, age of the work item, and age of the process or order.

An assignment plan may be created, such as by a supervisor, to map profiles to the end users whom he supervises such that work items may be assigned to appropriate end users. Work items may be wholly or partially manually assigned by a supervisor or self-assigned by the end user. Work items may also be auto-assigned using a wizard-like interface, for example, and can be triggered upon completion of another work item.

A supervisor may create several assignment plans and activate a particular assignment plan to set roles and priorities for his supervised end users. **FIG. 5** illustrates an example of an assignment plan 180 mapping profiles to end users. As shown, the assignment plan 180 can be represented by a matrix. Each end user is associated with one or more profiles and each profile is associated with one or more users.

Each end user preferably has an end user account on the order management system. The end user may log in to his account via the GUI. The user account specifies various field such as a user name, password, privilege, role, associated profile(s), supervisor, create time, and created by.

The end user may log into his account with a username and a password, for example. Preferably, the order management system requires each non-supervisory end user to be associated with at least one profile in order for the end user to log into his account. Upon logging into the account, all work items assigned to the end user may be loaded and displayed via the GUI. In addition, all appropriate work items which are active and have yet to be assigned to an end user may be loaded and displayed based on the user profile, the role specified by the work item, and the assignment plan, for example.

Work items may be assigned to the end users in various ways. In one embodiment, the end user may select, for example, "Get Next Work Item" option to retrieve the next priority work item(s) matching the end user's profile. Such an embodiment utilizes system-driven sequencing and prioritization.

Alternatively, the end user may select "Lookup Order" option by specifying certain order criteria, such as the identify of the client and/or customer and/or an order number, for example. The system may retrieve one or more orders that match the specified order criteria and, optionally, whose work items match the end user's role and profile. Such an

option may be utilized when the end user receives an e-mail, telephone, or facsimile from a client or customer requesting certain actions with respect to the order. The end user may then choose from the retrieved orders. In yet another alternative, an supervisor of the end user may manually assign work item(s) to the end user. The order management system preferably records all information entered, actions taken on work items, as well as the login event itself such as the user name, login time, and logout time.

When a work item is assigned to the end user, the end user may commit to the assigned work item by registering a commit time, i.e., the time when work is expected to be complete and/or work on again. The work item may require certain data to be captured prior to the work item being complete. The end user can optionally raise an exception against the work item and thus raise an exception flow. In addition, a work item assigned to an end user may be released prior to its completion so that it becomes available to be reassigned to another end user.

The work item may be escalated either automatically by the order management system based on time and/or other predefined criteria and/or manually by an end user assigned to the work item.

Work Item, Work Item Type, and Work Item Infrastructure

The work item infrastructure 136 contains meta data on work item, work item types, roles, profiles, assignment plans, data on the end users and their attributes, and the work item worklog, for example.

Work item types may be created and maintained. A work item type may be defined by the work item type name, description, priority, exception flag, role, escalation role, relative time, absolute time, input data, output data, and/or possible exceptions. Each work item type optionally has a generic exception work item type associated therewith to account for the occurrence of unanticipated exceptions. In one embodiment, the generic exception work item type requires description of the exception and requires that the exception be resolved manually. Any specific exception work item type may be added to the order management system for any recurring exceptions.

Work items of a particular work item type may be created when transitioning to a new state. The work item may be identified by the work item type, order, work item (if this is an exception, points to the normal work item when it is raised), time created, created by, start time (i.e., time from which relative timeout starts ticking), time assigned, assigned by, assigned user, commit time, committed by, time completed, time last touched, last touched by, relative time due (e.g., start time and relative time of work item type), absolute

time due, relative time expired, absolute time expired, status (e.g., active, inactive, suspended, cancelled, completed), severity (e.g., normal, minor, major).

For each work item, definition of one or more of several fields may be required. Examples of fields for a given work item include the work item name or identification, entry criteria and/or start time, work item description, possible results as determined by the user, and data needed corresponding to each result, such as data required for the order, for root cause analysis, and/or for reporting requirements, for example. In addition, a result determined by the system field may specify business rules for determining the result of the work item. An e-mail field corresponding to each result may specify whether an e-mail should be sent, to whom, and, if so, using which template. A worklog field may specify whether a worklog entry should be created and, if so, using which template. A role field specifies the applicable roles of the users who should or may be responsible for performing the work item actions. In addition, an escalation criteria and role field may optionally be defined for specifying the relative time from the start time of the work item to escalate the work item, the absolute time from the receipt of the order to escalate the work item, and to what role is the work item escalated.

Example of a Process for Fulfilling A Single Service Order in an Ideal Process Transitioning Only Through Normal States

FIG. 6 is a flow chart illustrating a process 600 for fulfilling a single service order utilizing the order management system in the absence of exceptions. At step 602, a service installation order is entered via the CFI and forwarded to the order management system GUI and causes the system to create an install order. If the order is successfully qualified by the order management system, the CFI may return an order ID to the customer. For example, upon successful qualification of the order, the appropriate business objects for this order are created and the order enters its initial state and the first work item required to process the order is created . Upon the entry of an order, the CFI preferably creates an order interface records to record and maintain all data required to fulfill order.

At step 604, the business objects of the order management system create entity beans. At step 606, the order management system sends a create order even to the business process engine and, in response at step 608, the business process engine creates an order business process objects and sends a create workflow event to the work item infrastructure.

At step 610, the work item infrastructure creates a workflow business object and sends a workflow created event to the business process engines. Then, at step 612, the

business process engine creates a workflow business process object and sends a create work item event to the work order infrastructure. Next, at step 614, the work infrastructure creates a work item business object and sends a work item created event to the business process engine and, at step 616, the business process engine creates a work item business process object.

At step 618, the work item infrastructure assigns the work item and updates the work item history. At step 620, the order information is retrieved and at step 622, upon action taken on the work item, entity beans are updated and the event is posted to the business process engine. Any e-mail deliveries and/or worklog updates are also performed here. Finally, at step 624,the update work item status event is posted. At this point, the work item history includes events such as the work item created, work item assigned, action taken, and work item status event posted.

Example: State Model/Business Process For A Service Installation Order

FIG. 7 is a state diagram of a state model/business process 700 illustrating the work items associated with an initial service installation order 702 in the absence of any exceptions. The work items associated with the initial service installation order 702 generally include a qualify order work item 704, an assign inventory work item 706, a place loop order work item 708, a receive FOC work item 710, a verify loop work item 712, a provision work item 714, a schedule install work item 716, and a deliver service work item 718.

The qualify order work item 704 is started or entered upon receipt of a service request order placed or entered via CFI/XML interface by a customer, for example. The qualify order work item 704 may be performed to verify the client's or service requester's address and the CO with the ILEC. Possible results of the qualify order work item 704 as determined by the system or the user include Success, Bad Address, and Bad CO. The assign inventory work item 706 is entered after the qualify order work item 704 is successfully completed and assigns a twisted pair to the client where the possible results as determined by the system include Success and Failure.

The place loop order work item 708 is entered after the assign inventory work item 706 is successfully completed and places a loop order where the possible result as determined by the user includes Success. The receive FOC work item 710 is entered after the place loop order work item 708 is successfully completed and the possible results as determined by the user include Success, Order Rejected by ILEC, Order Cancelled by

ILEC, No Facilities, Bad Facilities, TNF (Technically Not Feasible), Busy Pair, Bad Address, Bad CO, and Late FOC, for example.

The verify loop work item 712 may be entered after the FOC date is reached and is performed to verify that the loop may have been determined. The possible results as determined by the system include Successful test, Loop Not Delivered, and Failed Test.

The provision work item 714 is also entered after the assign inventory work item 706 is successfully completed and is performed to provision a private virtual circuit (PVC) from the pair to the remote customer. The possible results of the provision work item 714 as determined by the system include Success, No CO, No Card, No Circuit, and Provisioning Errors.

The schedule install work item 716 is entered after the receive FOC date and the provision work items 710, 714 are both successfully completed and is performed to schedule an installation via the workflow management system. The possible results of the schedule install work item 716 as determined by the system include Success and Failure.

Lastly, the deliver service work item 718 is entered after the schedule install work item is successfully completed and is performed in order to install and verify that service is up. The possible results of the schedule install work item 718 as determined by the system include Success, Close, Reschedule, TAC Resolution, Loop Problem, Lop Not Delivered, Customer Action Required, and End User Cancelled, for example.

If each of the required work items associated with the state model/business process 700 for the service installation order 702 is successfully completed, the business process completion state 720 is reached and the business process is complete.

Example: State Model/Business Process For Loop Not Delivered Due to Busy Pair Exception

FIG. 8 is a state diagram of a state model/business process 730 or a portion of a state model/business process illustrating the work items associated with a loop not delivered due to busy pair exception. The loop not delivered due to busy pair exception state may be entered upon determining that a Busy Pair is the result of the receive FOC work item (as shown in **FIG. 7**). The work items in the loop not delivered due to busy pair exception state model/business process 730 generally include a verify pair work item 732, a escalate disconnection work item 734, a mark pair busy work item 736, an allocate pair work item 738, a supplement loop order work item 740, a create new service work item 742, and a provision work item 744.

The verify pair work item 732 is entered when the ILEC indicates that the loop order is undeliverable due to a busy pair. Given the purchase order number (PON) and the ILEC circuit number, the verify pair work item 732 is performed to determine if the loop was ordered on the wrong pair, if the busy loop was ordered on the wrong pair, and/or if the busy loop is to be disconnected. The possible results of the verify pair work item 732 as determined by the system or the user may include incorrect pair ordered, loop order collision, and busy pair is disconnecting, each with a corresponding entry in the work item worklog. In addition, if the result is that the busy pair is disconnecting, the data that should be captured prior to completion of the verify pair work item 732 is the disconnection loop order number.

The escalate disconnection work item 734 is entered when the pair is busy because the ILEC has not disconnected the old loop. It is performed to instruct the ILEC to execute the disconnect order and to deliver the loop on the ordered pair with an escalated priority. The possible results as determined by the system or the user include concurrence by the ILEC and refusal by the ILEC, and the corresponding data that should be captured include the date of the escalation of the disconnection order and the date of the refusal by the ILEC, respectively.

The mark pair busy work item 736 is entered when the pair is determined to be genuinely busy and is performed to mark the pair as busy in the pair assignment database. The data that should be captured include PON and the ILEC circuit number of the busy pair. The possible results of the mark pair busy work item 736 as determined by the system or the user include pair marked busy and pair operation fails and the corresponding worklog entries include pair marked busy along with indication of the pair, PON, and ILEC circuit number, and pair operation failed, respectively.

The allocate pair work item 738 is entered upon when an old pair has been marked busy and is performed to assign a new pair to the order. The date and time of the assignment is preferably captured as data. The possible results as determined by the system or the user include pair assignment pairs and pair assigned with corresponding worklog entries of pair assignment failed and pair assigned along with pair and CO identifications, respectively.

The sup loop order work item 749 is entered when a pair has been allocated or it has been determined that the old pair was not correctly ordered. The sup loop order work item 749 is performed to supplement the existing loop order with the ILEC, specifying the assigned pair. The possible results as determined by the system or the user include sup

accepted and sup failed with corresponding worklog entries of sup accepted along with pair and CO identifications, and sup not accepted, respectively.

The create new service work item 742 is entered when a new pair has been assigned and is performed to create a new client service object using the assigned pair and stack the new client service object on the service stack. The possible results as determined by the system or the user include new service created and service create failed.

The provision work item 744 is entered when a new service has been created and stacked and is performed to provision the new service. Preferably, the auto-provisioning service removes the existing provisioning before the re-provisioning. The possible results as determined by the system include provisioning succeeded and provisioning failed. The data which should be captured includes the date of the provision or the provision failure and the worklog entry includes whether the provisioning succeeded or failed and the service identification.

As is evident, work items for numerous other business processes for orders and exceptions may be defined and implemented and/or modified in the order management system. Other examples of business processes for orders mentioned above include cancel client installation order, service change order, disconnect order, provider (such as provider of Internet Service) switch order, cancel disconnect order, cancel change order, change service on installation order in progress order,

Recording History of Work Items and Generating Reports

As discussed above, various data regarding a work item or an order having associated work items, for example, may be recorded and its history updated and maintained. For example, a worklog and/or work item history may be created and maintained to record creations, status, changes, deletions, etc. for each work item.

A worklog generally refers to the operational transitions of the state model, i.e., a process or order level log. Not all work items causes creation of an entry in a worklog. The worklog is typically defined by meta data and describes transitions at a higher level than the work item history. Examples of the worklog entries include received order, verified order, FOC date received, and provision service exception.

In contrast, work item history is associated with a work item and can be very detailed. Work item history may contain information such as system administration statistics, for example. Exemplary work item data includes when and to whom the work item is assigned, time last touched, last touched by, actions performed, release or

completion time, exceptions, and any communication such as e-mail delivered. In general work item history contains the data to drive the metrics to be generated.

Various reports may be generated from the data collected and stored. Reports may be generated using filters and the resulting reports can be sorted according to one or more criteria such as customer, customer type, ILEC, region, CO, order type, order states, order number, circuit number, service priority, work item type, status, severity, service, dates, assigned end user, end user role.

FIG. 9A illustrates an example of a GUI 900 for specifying the criteria and selecting a specific type of queue management report to be generated from the data maintained by the order management system. **FIG. 9B** illustrates an example of a workflow summary report generated using the data maintained by the order management system.

In the example shown, the GUI 900 provides for specification of role, administrative region, region, CO, ILEC, customer, product type, and/or report type. Although not shown, filters made be utilized to include and/or exclude specified work item type, work flow, order type, and/or customer's service provider, for example. In addition, the resultant report can optionally be grouped according to work item, work item type, workflow, order type, and/or customer's service provider for example.

FIG. 9B illustrates an example of a workflow summary report 902 that summarizes the workflow for one or more roles that meet the selection criteria specified via the GUI. The workflow summary report 902 facilitates in managing the overall workflow and in spotting exceptions within operational areas, for example. In the example shown in **FIG. 9B**, the workflow summary report 902 provides the number of open work items (action or inactive), the number of work items with escalation, the number of work items with exception(s) (i.e. inactive work items), the number of work items that are unassigned. In addition, the workflow summary report 902 provides order velocity metrics by day and/or week. More particularly, the order velocity metrics may include the number of work items that were opened in a given time period, the number of number of work items that were closed or completed in a given time period, the number of number of work items that were touched in a given time period, as well as the number of orders falling within the specified days ARO (after receipt of order).

FIGS. 9A and **9B** are merely illustrative of one type of workflow summary report. Numerous other reports may also be generated depending upon the application and the desired statistics or the desired monitoring. Examples of other data, analysis, or information that may be presented in such reports include the number of orders of a given

type, the number of orders on track, the number of orders off track, the number of orders in a given state, the number and average/minimum/maximum time for specific work item types, the number of work items or exceptions of a particular type sorted according to the amount of time worked on or frequency of occurrence, the processes in which exceptions occur, work load distribution, the end users to whom work is assigned, and/or end users who has completed which work.

Examples of other types of reports include workflow summary (client order queues) report, work item summary report, work item list, customer/client order report, workflow process analysis report, work item summary analysis report, work item detail analysis report, resources productivity report, workload forecast report, touch report, queue management report, jeopardy management report, resource load management report, worker effectiveness and efficiency report, exception management report, and process analysis and design report. The criteria specified for the reports may exclude certain items in order to generate more refined metrics.

The order management system may also facilitate fulfillment of customer care requirements such as order status inquiry regarding a specified order, whether the order is active such as when the order is in process, inactive such as when the order cannot be completed and no further action is to be taken, or complete such as when all required work items have been performed. The order management system may perform lookups of a client order via the GUI according to the PON, ILEC circuit, CLEC circuit, client name, client address, client phone, customer, and/or date, for example. The order management system may locate the order and retrieve all, only active work items, completed work items, and/or exceptions associated with any of the work items, for example.

As described above, the process fulfillment systems and methods using distributed workflow architecture described herein is generally separated into three components: business objects, business process definition, and meta data in the form of a work item infrastructure module or server. Such an organizational architecture limits the impact from changes that may need to be made to the fulfillment process.

For example, a change to the fulfillment process may require only changes to the business process. Pure business process changes are typically implemented with modifications to the states or state flow model using existing work items and business objects. Pure business process modifications may include, for example, removing or adding a step in the state model and/or the reordering of state dependencies. For example, the business process may initially require that prior to performing loop testing, the CLEC

must confirm with the ILEC that the ILEC work was completed after the FOC date is reached. The business process may be modified to omit the step of confirming completion of the work with the ILEC and proceed directly to loop testing. In such a process change-only case, the appropriate state model need to be changed and no other change needs to be made to the work items, the work item infrastructure, and various other components of the system.

In addition, certain modifications to the fulfillment process may only require changes to the meta data of the work item infrastructure, typically when there is no change in the data being collected. Examples of process fulfillment modifications requiring only meta data-only changes include changes to escalation, the roles, and the start time of a work item.

Similarly, where there is a change in the business logic, i.e., the business objects, no change to the business process engines or the work item infrastructure may be necessary. An example of such a case is where it is change the business objects such that certain data is stored in multiple places in multiple objects.

It may be desirable to invoke a different work item for the same or similar task depending upon the circumstances under which the work item is invoked. For example, receive FOC date may occur under various circumstances such as in response to an initial loop order, a sup-order, or after an exception. A different work item may be created under each different circumstance such that the work items may have different start time, different escalations, and/or different roles, for example. For such a modification to the system, the business process may be modified such that a different work item is invoked depending upon the present state of the process, for example.

In another scenario, modifications to the process may require changes to two or three of the components of the process fulfillment system. For example, it may be desirable to modify the system to store additional information when certain work item actions are taken, such as the storing of loop test results. In such a case, the work item infrastructure may need to be modified such that when the action is taken the desired data is captured and, in addition, the business objects may need to be modified in order to store the desired information captures by the work item infrastructure.

Exemplary Computer System

FIGS. 10 and 11 illustrate a schematic and a block diagram, respectively, of an example of a general purpose computer system 1000 suitable for executing software programs that implement the methods and processes described herein. The architecture

and configuration of the computer system 1000 shown and described herein are merely illustrative and other computer system architectures and configurations may also be utilized.

The illustrative computer system 1000 includes a display 1003, a screen 1005, a cabinet 1007, a keyboard 1009, and a mouse 1011. The mouse 1011 can have one or more buttons for interacting with a GUI (graphical user interface) that may be displayed on the screen 1005. The cabinet 1007 typically house one or more drives to read a computer readable storage medium 1015, system memory 1053, and a hard drive 10155, any combination of which can be utilized to store and/or retrieve software programs incorporating computer codes that implement the methods and processes described herein and/or data for use with the software programs, for example. Examples of computer or program code include machine code, as produced, for example, by a compiler, or files containing higher level code that may be executed using an interpreter.

Computer readable media may store program code for performing various computer-implemented operations and may be encompassed as computer storage products. Although a CD-ROM and a floppy disk 1015 are shown as exemplary computer readable storage media readable by a corresponding CD-ROM or floppy disk drive 1013, any other combination of computer readable storage media can be utilized. Computer readable medium typically refers to any data storage device that can store data readable by a computer system. Examples of computer readable storage media include tape, flash memory, system memory, and hard drive may alternatively or additionally be utilized. Computer readable storage media may be categorized as magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROM disks; magneto-optical media such as floptical disks; and specially configured hardware devices such as application-specific integrated circuits (ASICs), programmable logic devices (PLDs), and ROM and RAM devices. Further, computer readable storage medium may also encompass data signals embodied in a carrier wave, such as the data signals embodied in a carrier wave carried in a network. Such a network may be an intranet within a corporate or other environment, the Internet, or any network of a plurality of coupled computers such that the computer readable code may be stored and executed in a distributed fashion.

Computer system 1000 comprises various subsystems. The subsystems of the computer system 1000may generally include a microprocessor 1051, system memory 1053, fixed storage 1055 (such as a hard drive), removable storage 1057 (such as a CD-ROM drive), display adapter 1059, sound card 1061, transducers 1063 (such as speakers and microphones), network interface 1065, and/or scanner interface 1067.

The microprocessor subsystem 1051 is also referred to as a CPU (central processing unit). The CPU 1051 can be implemented by a single-chip processor or by multiple processors. The CPU 1051 is a general purpose digital processor which controls the operation of the computer system 1000. Using instructions retrieved from memory, the CPU 1051 controls the reception and manipulation of input data as well as the output and display of data on output devices.

The network interface 1065 allows CPU 1051 to be coupled to another computer, computer network, or telecommunications network using a network connection. The CPU 1051 may receive and/or send information via the network interface 1065. Such information may include data objects, program instruction, output information destined to another network. An interface card or similar device and appropriate software implemented by CPU 1051 can be used to connect the computer system 1000 to an external network and transfer data according to standard protocols. In other words, methods and processes described herein may be executed solely upon CPU 1051 and/or may be performed across a network such as the Internet, intranet networks, or LANs (local area networks), in conjunction with a remote CPU that shares a portion of the processing. Additional mass storage devices (not shown) may also be connected to CPU 1051 via the network interface 1065.

The subsystems described herein are merely illustrative of the subsystems of a typical computer system and any other suitable combination of subsystems may be implemented and utilized. For example, another computer system may also include a cache memory and/or additional processors 1051, such as in a multi-processor computer system.

The computer system 1000 also includes a system bus 1069. However, the specific buses shown are merely illustrative of any interconnection scheme serving to link the various subsystems. For example, a local bus can be utilized to connect the central processor to the system memory and display adapter.

The computer system 1000 as shown in FIGS. 10 and 11 may be illustrative of the computer system of the service provider, the client, and/or the end-user or customer.

While the above is a complete description of preferred embodiments of the invention, various alternatives, modifications, and equivalents can be used. It should be evident that the invention is equally applicable by making appropriate modifications to the embodiments described above. Therefore, the above description should not be taken as limiting the scope of the invention that is defined by the appended claims along with their full scope of equivalents.

CLAIMS

What is claimed is:

1. A distributed workflow system for fulfillment of a process, comprising:

a business objects module defining business rules relating to the process;

a business processes module defining state models of the process, each state model defining states and state transitions; and

a work item infrastructure module containing work item business objects and work item meta data,

wherein said business process module is configured to cause generation of work items in said work item infrastructure on which work item actions can be performed and wherein performance of work items actions causes transition within the state model in said business processes module and causes updating of the business objects module.

2. The distributed workflow system for process fulfillment of claim 1, further comprising a remote method invocation bus for invoking business methods and for enabling an external system to activate components in said business objects module.

3. The distributed workflow system for process fulfillment of claim 1, further comprising an events channel, wherein said business objects, business processes, and work item infrastructure modules are adapted to post events of said process on said events channel.

4. The distributed workflow system for process fulfillment of claim 3, wherein each of said business processes module and said work item infrastructure module selectively subscribes to events on said events channel.

5. The distributed workflow system for process fulfillment of claim 1, wherein said business objects module comprises at least one entity bean and at least one session bean, said at least one session bean for invoking business logic and for enforcing said business rules.

6. The distributed workflow system for process fulfillment of claim 5, wherein said business objects module comprises a meta session bean, an order session bean, a work items session bean, and a user session bean.

7. The distributed workflow system for process fulfillment of claim 1, wherein said business processes module includes business process objects for creating said work items in said work item infrastructure.

8. The distributed workflow system for process fulfillment of claim 1, wherein said business objects module further contains data relating to roles and profiles assigned to end users of the system and wherein assignment of each work item action to the end user is based upon the role and profile specified by the work item meta data and the role and profile of the end user.

9. The distributed workflow system for process fulfillment of claim 1, further comprising a data layer for storing data obtained by at least one of said business objects, business processes, and work item infrastructure modules.

10. The distributed workflow system for process fulfillment of claim 9, further comprising a reporting module for processing data stored in said data layer and for report generation.

11. A method for process fulfillment using a distributed workflow architecture, comprising:

generating work items by a business process engine on a work item infrastructure module, said work item infrastructure module containing work item business objects and work item meta data;

transitioning within a state model of the process defined in said business processes module upon performance of a work item action associated with one of said generated work items, said state model defining states and state transitions;

updating a business objects module upon performance of the work item action, said business objects module defining business rules relating to the process; and

repeating said generating, transitions, and updating for each state of the state model until disposition of the process.

12. The method for process fulfillment using a distributed workflow architecture according to claim 11, further comprising invoking business methods on a remote method invocation bus and enabling an external system to activate components in said business objects module.

13. The method for process fulfillment using a distributed workflow architecture according to claim 11, further comprising posting of an event of the process to an events channel by at least one of said business objects, business processes, and work item infrastructure modules.

14. The method for process fulfillment using a distributed workflow architecture according to claim 13, further comprising selectively receiving events on said events channel by one of said business processes module and said work item infrastructure module based upon events subscription of the receiving module.

15. The method for process fulfillment using a distributed workflow architecture according to claim 11, further comprising invoking business logic and enforcing business rules by at least one session bean of the business objects module.

16. The method for process fulfillment using a distributed workflow architecture according to claim 11, wherein said generating the work items is performed by business process objects of the business processes module.

17. The method for process fulfillment using a distributed workflow architecture according to claim 11, further comprising assigning a work item action to an end user of the system according to role and profile of the end user by the business objects module.

18. The method for process fulfillment using a distributed workflow architecture according to claim 11, further comprising storing data in a data layer, said data obtained by at least one of said business objects, business processes, and work item infrastructure modules.

19. The method for process fulfillment using a distributed workflow architecture according to claim 18, further comprising processing data stored in said data layer and generating report therefrom by a reporting module.

20. A process fulfillment system, comprising:

means for defining business rules relating to the process to be fulfilled;

means for defining state models of the process, each state model defining states and state transitions; and

means for storing work item business objects and work item meta data,

wherein said means for defining state models is configured to cause generation of work items in said means for storing on which work item actions can be performed and wherein performance of said work items actions causes transition within the state model and causes updating of said means for defining business rules.

21. The process fulfillment system according to claim 20, further comprising means for invoking business methods and for enabling an external system to activate components in said means for defining business rules.

22. The process fulfillment system according to claim 20, further comprising means for posting events by at least one of said means for defining business rules, means for defining state models, and means for storing.

23. The process fulfillment system according to claim 22, further comprising means for selectively subscribing to events on said posting means by said state models defining means and by said storing means.

24. The process fulfillment system according to claim 20, wherein further comprising means for assigning work item action to an end user according to a role and profile of specified by the work item action and according to the role and profile of the end user.

25. The process fulfillment system according to claim 20, further comprising means for storing data obtained by at least one of said storing means, said state models defining means, and said business rules defining means.

26. The process fulfillment system according to claim 25, further comprising means for processing data stored in said data storing means and for generating a report therefrom.

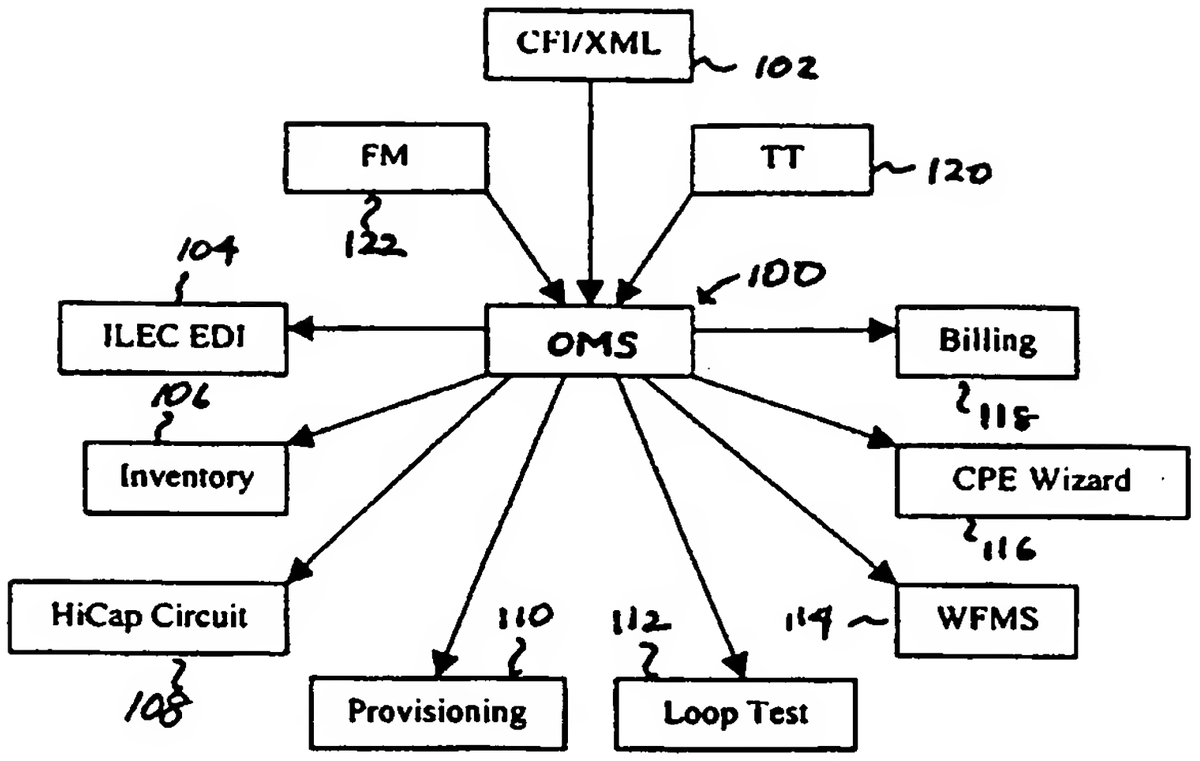

CFI/XML
102
FM
122
TT
120
104
100
ILEC EDI
OMS
Billing
106
118
Inventory
CPE Wizard
116
HiCap Circuit
110
112
114
WFMS
108
Provisioning
Loop Test

FIG. 1

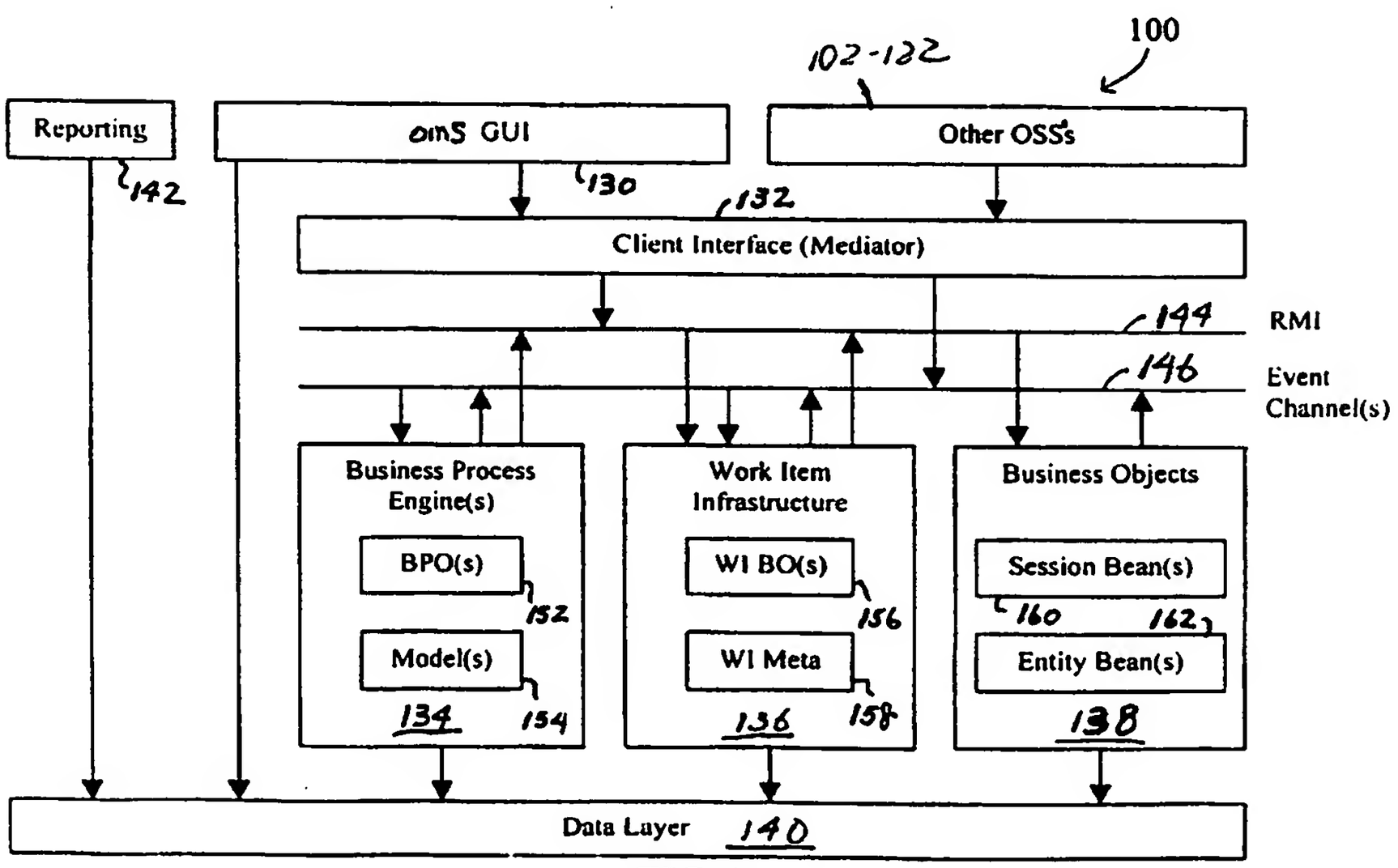

100
102-122
Reporting
142
OMS GUI
130
Other OSS's
132
Client Interface (Mediator)
144
RMI
146
Event Channel(s)
Business Process Engine(s)
Work Item Infrastructure
Business Objects
BPO(s)
152
WI BO(s)
156
Session Bean(s)
160
162
Model(s)
154
WI Meta
158
Entity Bean(s)
134
136
138
Data Layer
140

FIG. 2

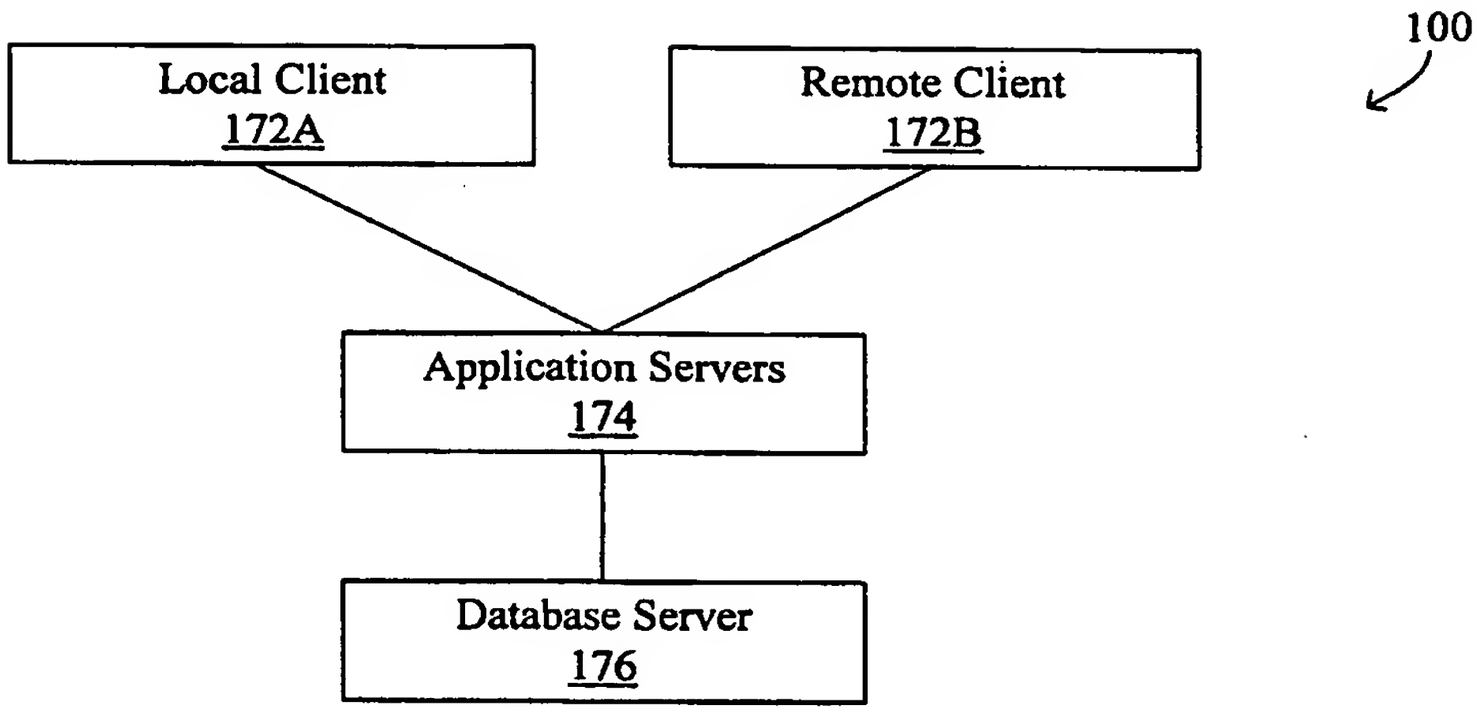

100
Local Client
172A
Remote Client
172B
Application Servers
174
Database Server
176

FIG. 3

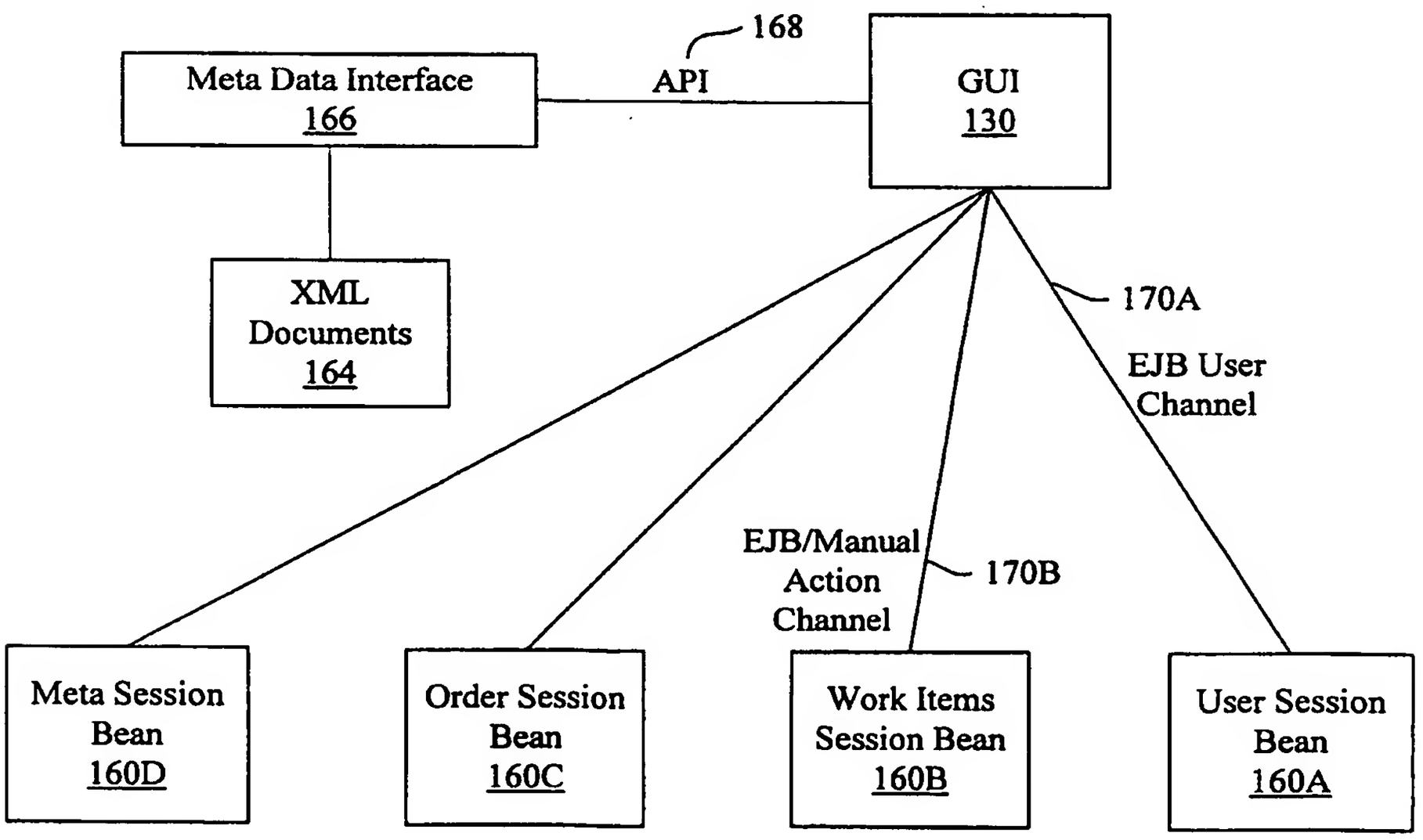

168
Meta Data Interface
166
API
GUI
130
XML
Documents
164
170A
EJB User
Channel
EJB/Manual
Action
Channel
170B
Meta Session
Bean
160D
Order Session
Bean
160C
Work Items
Session Bean
160B
User Session
Bean
160A

FIG. 4

180

ASSIGNMENT PLAN FOR SUPERVISOR Z								
	Personnel/End Users Under Supervisor Z							
Profiles		1	2	3	4	5	6	7
	A		X		X	X		
	B			X		X		X
	C		X				X	X
	D	X						

FIG. 5

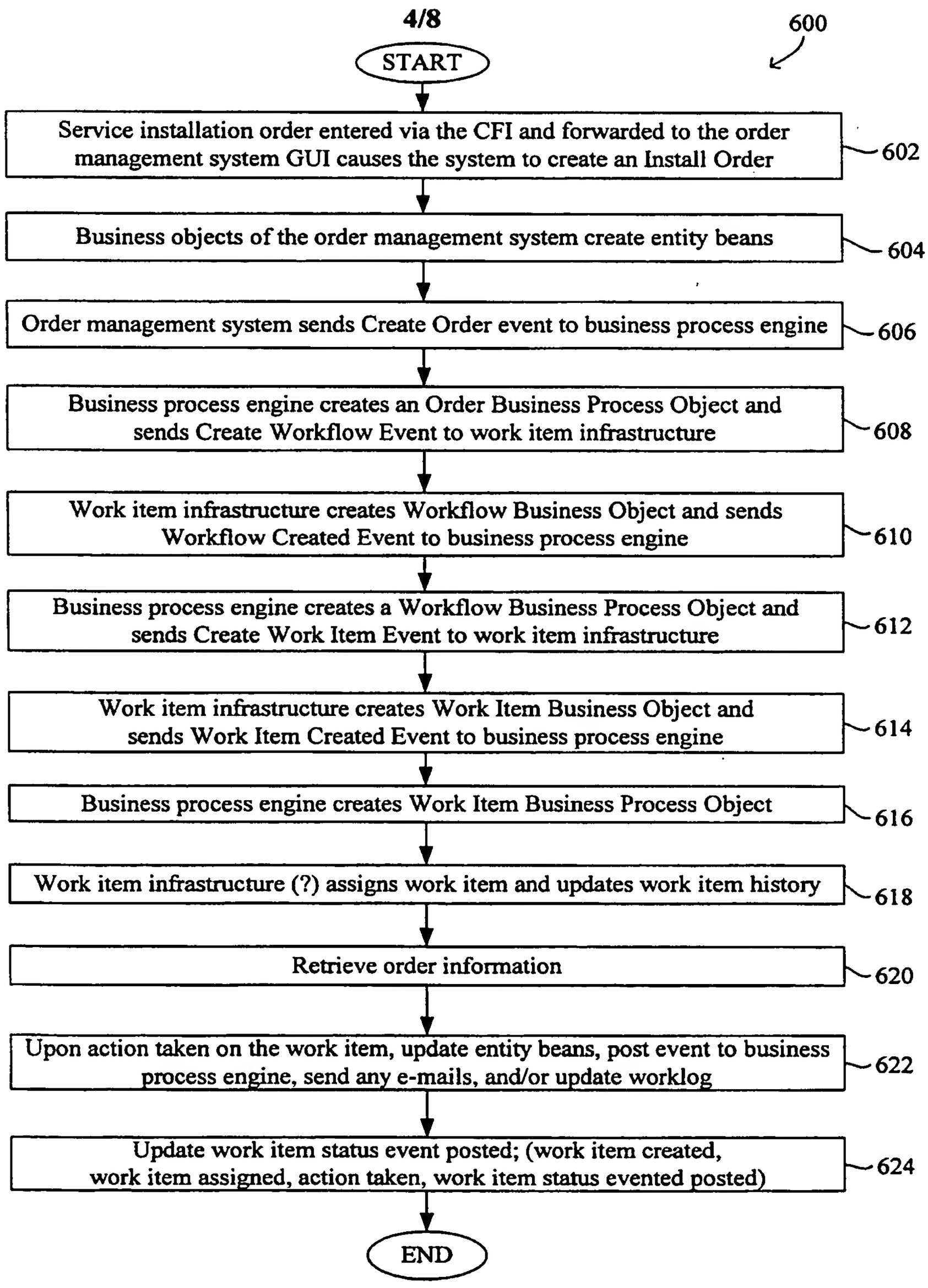

600
START
Service installation order entered via the CFI and forwarded to the order management system GUI causes the system to create an Install Order
602
Business objects of the order management system create entity beans
604
Order management system sends Create Order event to business process engine
606
Business process engine creates an Order Business Process Object and sends Create Workflow Event to work item infrastructure
608
Work item infrastructure creates Workflow Business Object and sends Workflow Created Event to business process engine
610
Business process engine creates a Workflow Business Process Object and sends Create Work Item Event to work item infrastructure
612
Work item infrastructure creates Work Item Business Object and sends Work Item Created Event to business process engine
614
Business process engine creates Work Item Business Process Object
616
Work item infrastructure (?) assigns work item and updates work item history
618
Retrieve order information
620
Upon action taken on the work item, update entity beans, post event to business process engine, send any e-mails, and/or update worklog
622
Update work item status event posted; (work item created, work item assigned, action taken, work item status evented posted)
624
END

FIG. 6

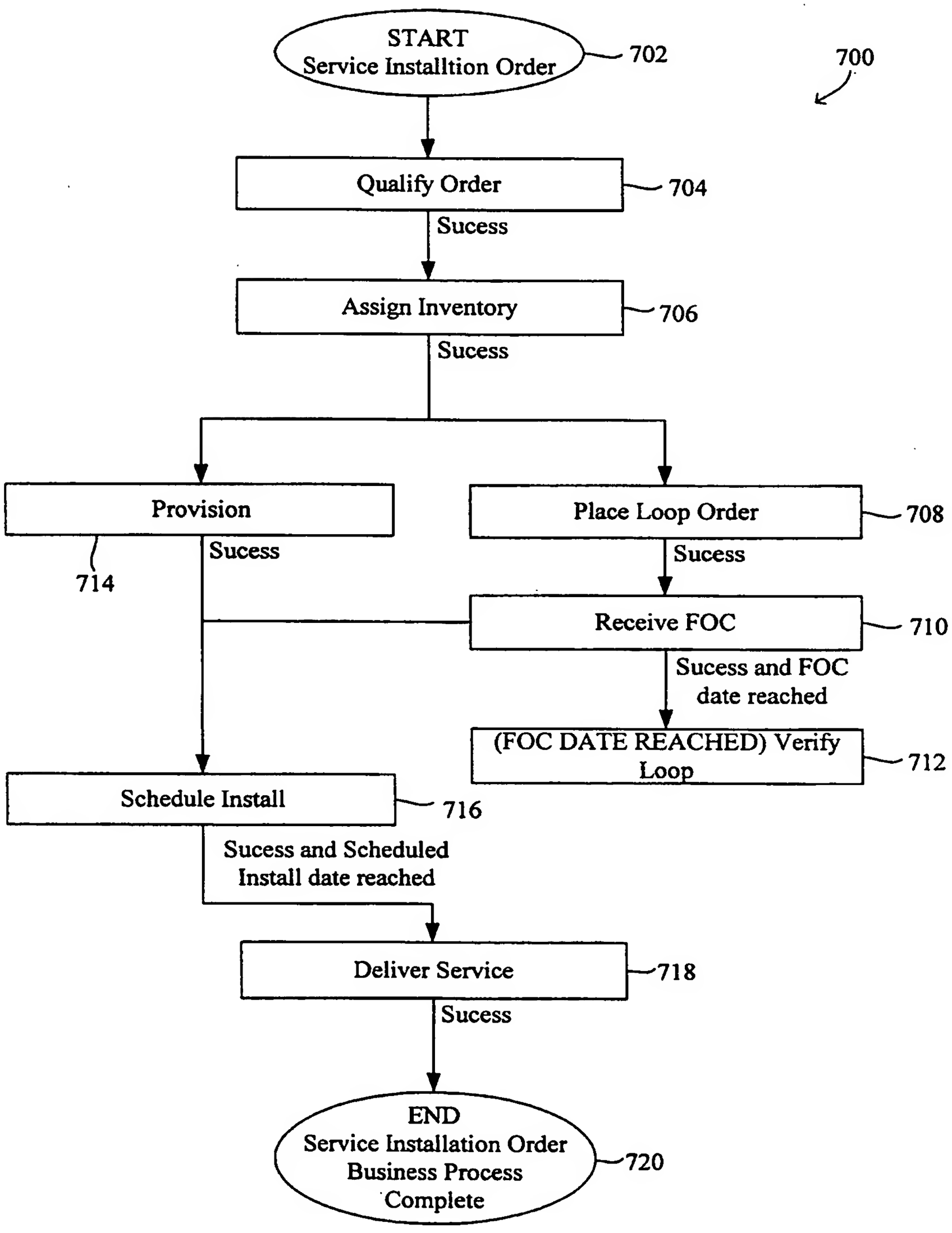

START
Service Installtion Order
702
700
Qualify Order
704
Sucess
Assign Inventory
706
Sucess
Provision
Sucess
714
Place Loop Order
708
Sucess
Receive FOC
710
Sucess and FOC
date reached
(FOC DATE REACHED) Verify
Loop
712
Schedule Install
716
Sucess and Scheduled
Install date reached
Deliver Service
718
Sucess
END
Service Installation Order
Business Process
Complete
720

FIG. 7

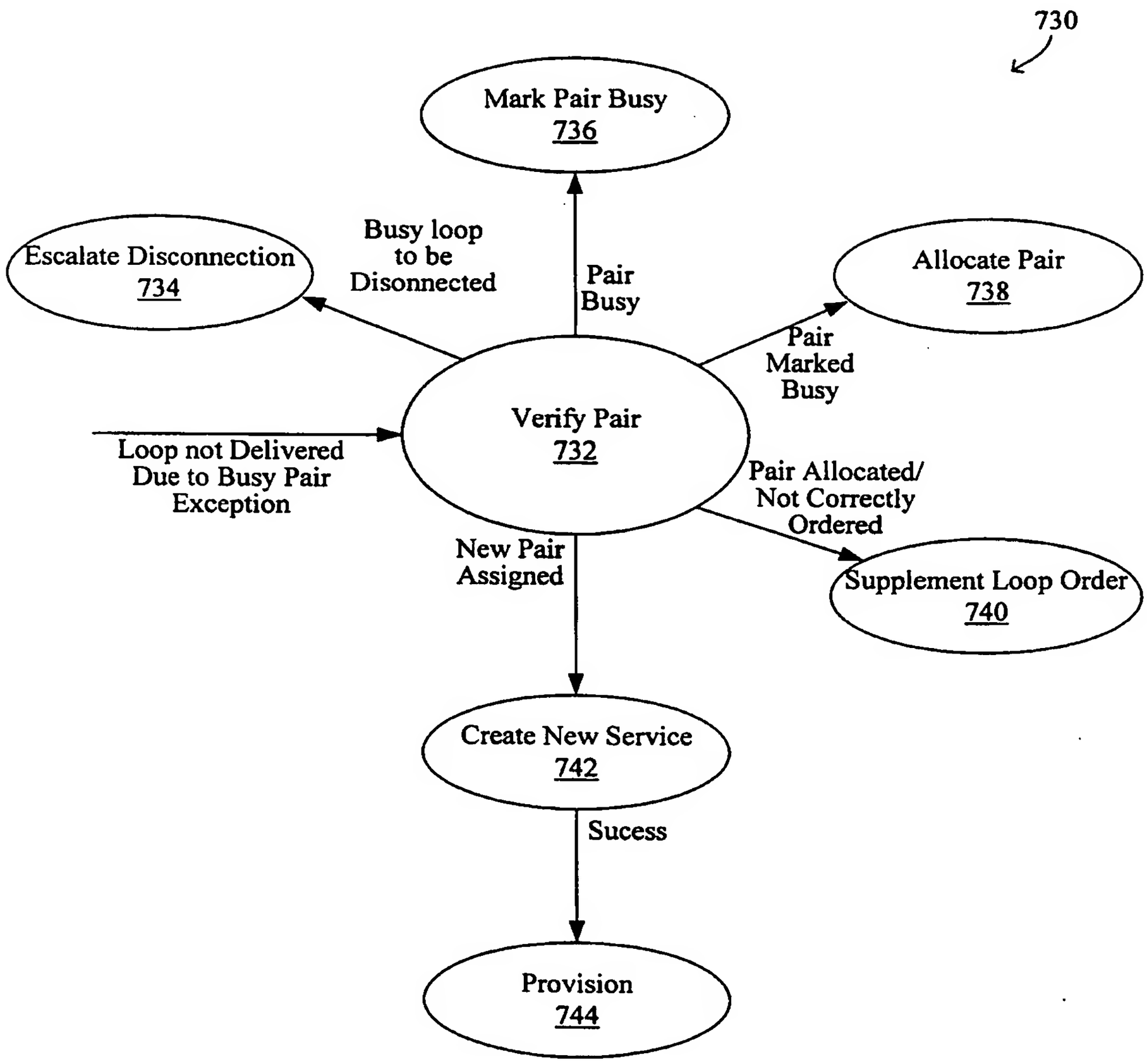

730
Mark Pair Busy
736
Escalate Disconnection
734
Busy loop
to be
Disonnected
Pair
Busy
Allocate Pair
738
Pair
Marked
Busy
Verify Pair
732
Loop not Delivered
Due to Busy Pair
Exception
Pair Allocated/
Not Correctly
Ordered
New Pair
Assigned
Supplement Loop Order
740
Create New Service
742
Sucess
Provision
744

FIG. 8

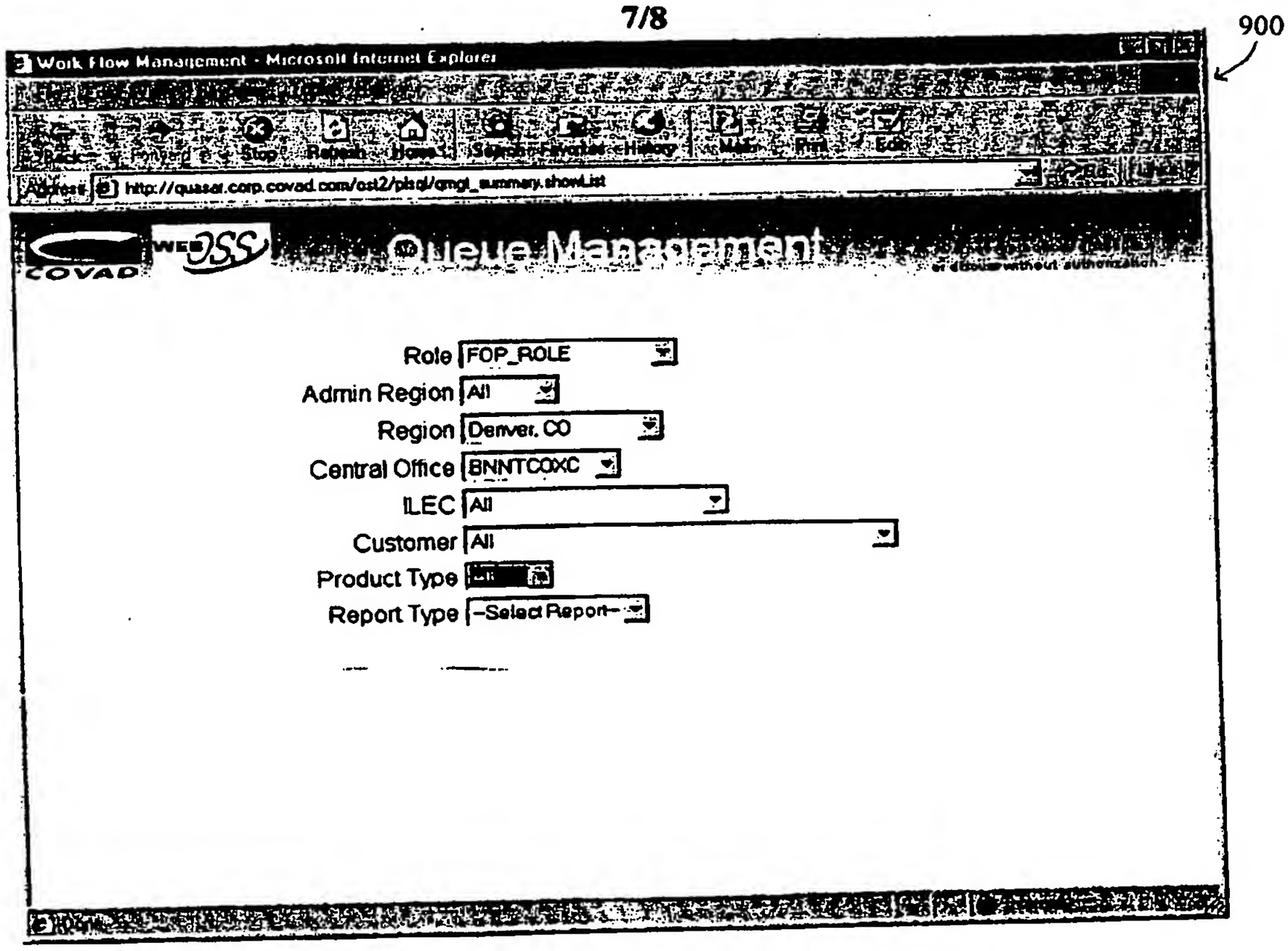

900
Work Flow Management - Microsoft Internet Explorer
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Address
http://quasar.corp.covad.com/osi2/plsql/qmgt_summary.showList
COVAD
WEB OSS
Queue Management
Role FOP_ROLE
Admin Region All
Region Denver, CO
Central Office BNNTCOXC
ILEC All
Customer All
Product Type
Report Type -Select Report-

FIG. 9A

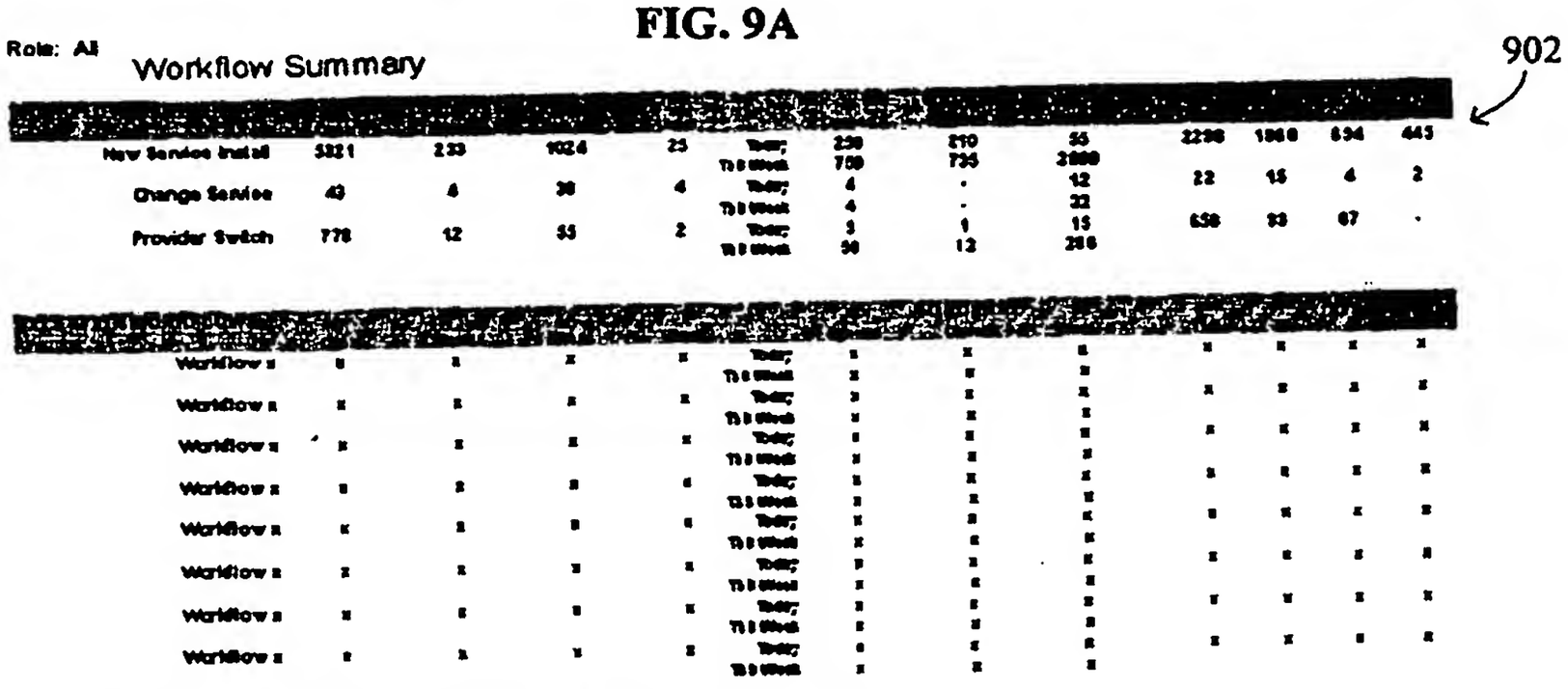

902
Role: All
Workflow Summary
New Service Install 3321 233 1024 25 Today 239 510 55 2298 1068 634 443
This Week 709 735 2889
Change Service 49 4 38 4 Today 4 - 12 22 15 4 2
This Week 4 - 22
Provider Switch 778 12 55 2 Today 3 1 15 658 33 67 -
This Week 36 12 286
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
Workflow x x x x x Today; This Week x x x x x x x x
All Roles, All Admin Regions, San Francisco, CA, all Cos, All Customers, All ILECS, all Product Types

FIG. 9B

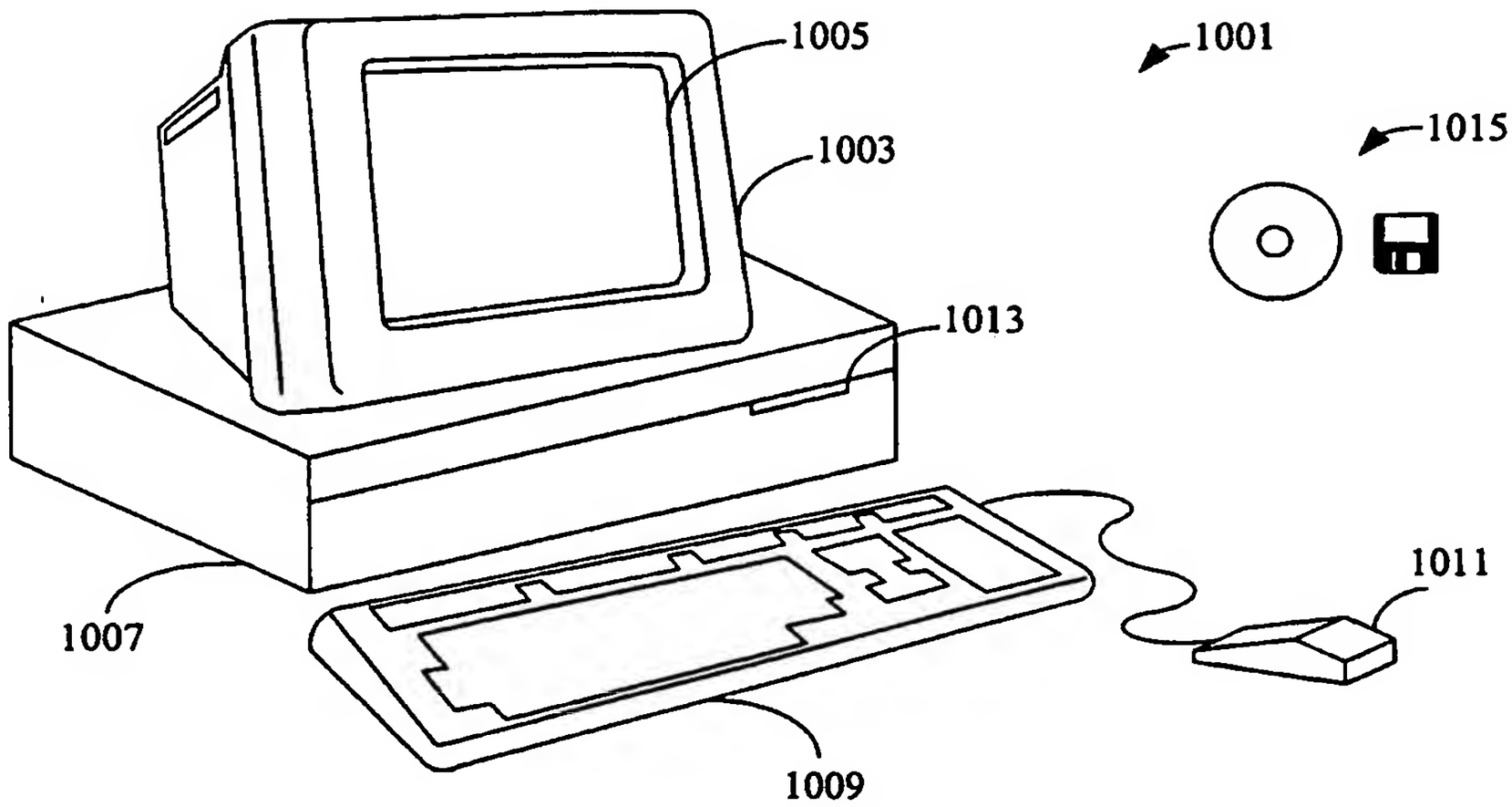

1005
1001
1015
1003
1013
1011
1007
1009

FIG. 10

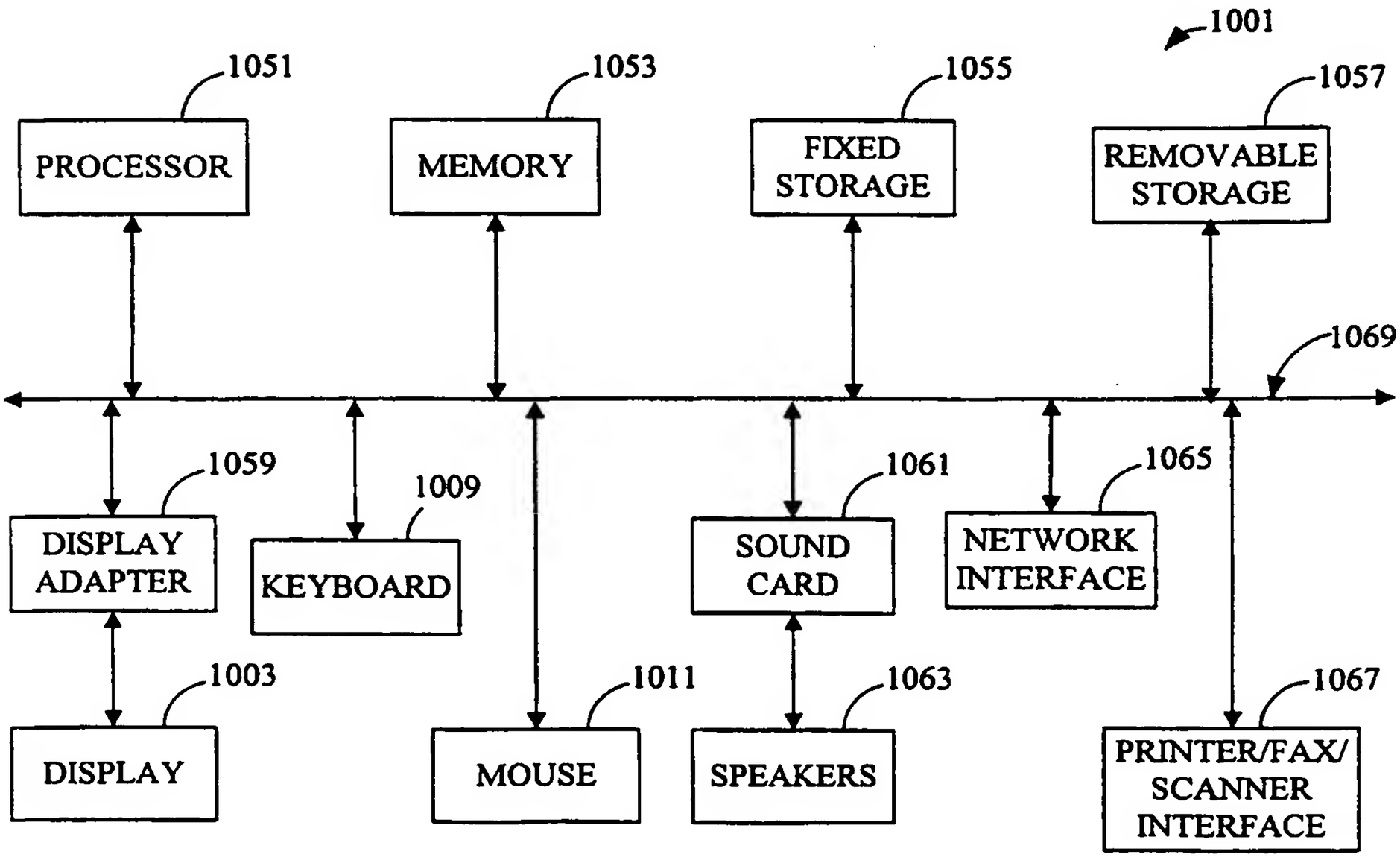

1001
1051
1053
1055
1057
PROCESSOR
MEMORY
FIXED STORAGE
REMOVABLE STORAGE
1069
1059
1009
1061
1065
DISPLAY ADAPTER
KEYBOARD
SOUND CARD
NETWORK INTERFACE
1003
1011
1063
1067
DISPLAY
MOUSE
SPEAKERS
PRINTER/FAX/ SCANNER INTERFACE

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/17921

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/16, 15/82, 17/60
US CL :705/26, 14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/26, 14

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, DIALOG

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,485,369 A (NICHOLLS et al) 16 January 1996, the ABSTRACT.	1-26
X	US 4,734,858 A (SCHLAFLY) 29 March 1988, the ABSTRACT.	1-26
X	US 5,732,400 A (MANDLER et al) 24 March 1998, the ABSTRACT.	1-26
X	US 5,909,492 A (PAYNE et al) 01 June 1999, the ABSTRACT.	1-26

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 24 SEPTEMBER 2000	Date of mailing of the international search report 06 OCT 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer JAMES P. TRAMMELL Telephone No. (703) 305-9768

Form PCT/ISA/210 (second sheet) (July 1998)★